Nested experimental design with *edgeR*, *limma* and *Dream*

CompBio/ML seminar, FMI

Michal Kloc

Bentires-Alj Lab, University Hospital Basel, Switzerland, March 7, 2023

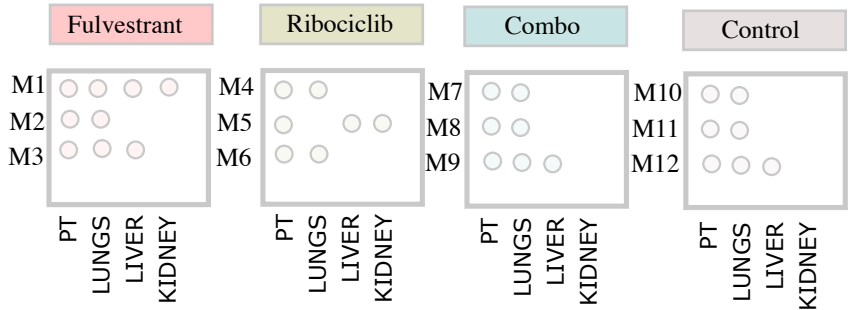
Outline

1. Experimental setting
2. Fixed-effect model approach
   challenges and solutions (*edgeR*)
3. Mixed-effect model approach (*limma* and *Dream*)
4. Summary

Useful paper:
A guide to creating design matrices for gene expression experiments, C.W. Law et al., 2020,
`https://f1000research.com/articles/9-1444/v1`

Experimental setting

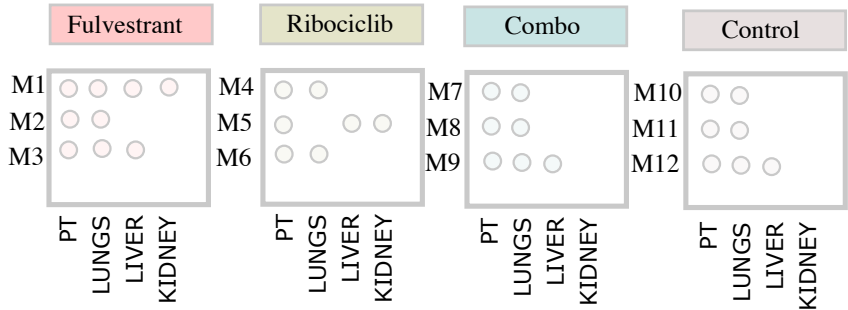
patient material (initial tumor) grown in mice: patient-derived xenographs (PDXs)



covariates: *treatment*, *site*, *mouse* $\leftrightarrow$ interaction and paired structure

Experimental setting

patient material (initial tumor) grown in mice: patient-derived xenographs (PDXs)

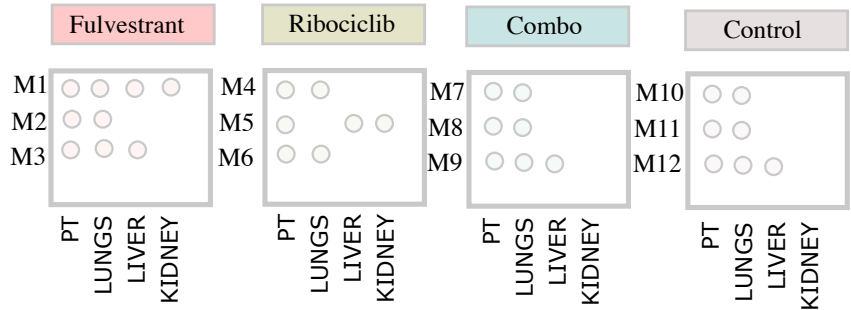


covariates: *treatment*, *site*, *mouse* $\leftrightarrow$ interaction and paired structure

dsg = model.matrix($\sim$ treatment.site + mouse)

Experimental setting

patient material (initial tumor) grown in mice: patient-derived xenographs (PDXs)



covariates: *treatment*, *site*, *mouse* $\leftrightarrow$ interaction and paired structure

dsg = model.matrix($\sim$ treatment.site + mouse)

FAILS! Design matrix would not have full rank (redundant covariates!!)

edgeR approach

- remove redundancy $\leftrightarrow$ treatments can be reconstructed from mouse covariates

```
#between and within subject comparisons
dsg.HCI013 <- model.matrix(~ 0 + mouse, data = dge.HCI013$samples)  #start with mouse effect
#site specific treatments, except PT, this is the reference
#table(dge.HCI013$samples$SampleGroup,dge.HCI013$samples$site)
Control.LUNGS <- dge.HCI013$samples$SampleGroup == "Control" & dge.HCI013$samples$site == "LUNGS"
Ribo.LUNGS <- dge.HCI013$samples$SampleGroup == "Ribo" & dge.HCI013$samples$site == "LUNGS"
Combo.LIVER <- dge.HCI013$samples$SampleGroup == "Combo" & dge.HCI013$samples$site == "LIVER"
Combo.LUNGS <- dge.HCI013$samples$SampleGroup == "Combo" & dge.HCI013$samples$site == "LUNGS"
Combo.KIDNEY <- dge.HCI013$samples$SampleGroup == "Combo" & dge.HCI013$samples$site == "KIDNEY"
Fulves.LUNGS <- dge.HCI013$samples$SampleGroup == "Fulves" & dge.HCI013$samples$site == "LUNGS"
Fulves.LIVER <- dge.HCI013$samples$SampleGroup == "Fulves" & dge.HCI013$samples$site == "LIVER"
Fulves.KIDNEY <- dge.HCI013$samples$SampleGroup == "Fulves" & dge.HCI013$samples$site == "KIDNEY"

dsg.HCI013 <- cbind(dsg.HCI013,Control.LUNGS,Ribo.LUNGS,Combo.LIVER,Combo.LUNGS,Combo.KIDNEY,Fulves.LUNGS,Fulves.LIVER,Fulves.KIDNEY)
dsg.HCI013 <- dsg.HCI013[, colSums(dsg.HCI013) > 0] # remove non-observable coefficients
```

edgeR approach

- remove redundancy $\leftrightarrow$ treatments can be reconstructed from mouse covariates

```
#between and within subject comparisons
dsg.HCI013 <- model.matrix(~ 0 + mouse, data = dge.HCI013$samples)  #start with mouse effect
#site specific treatments, except PT, this is the reference
#table(dge.HCI013$samples$SampleGroup,dge.HCI013$samples$site)
Control.LUNGS <- dge.HCI013$samples$SampleGroup == "Control" & dge.HCI013$samples$site == "LUNGS"
Ribo.LUNGS <- dge.HCI013$samples$SampleGroup == "Ribo" & dge.HCI013$samples$site == "LUNGS"
Combo.LIVER <- dge.HCI013$samples$SampleGroup == "Combo" & dge.HCI013$samples$site == "LIVER"
Combo.LUNGS <- dge.HCI013$samples$SampleGroup == "Combo" & dge.HCI013$samples$site == "LUNGS"
Combo.KIDNEY <- dge.HCI013$samples$SampleGroup == "Combo" & dge.HCI013$samples$site == "KIDNEY"
Fulves.LUNGS <- dge.HCI013$samples$SampleGroup == "Fulves" & dge.HCI013$samples$site == "LUNGS"
Fulves.LIVER <- dge.HCI013$samples$SampleGroup == "Fulves" & dge.HCI013$samples$site == "LIVER"
Fulves.KIDNEY <- dge.HCI013$samples$SampleGroup == "Fulves" & dge.HCI013$samples$site == "KIDNEY"

dsg.HCI013 <- cbind(dsg.HCI013,Control.LUNGS,Ribo.LUNGS,Combo.LIVER,Combo.LUNGS,Combo.KIDNEY,Fulves.LUNGS,Fulves.LIVER,Fulves.KIDNEY)
dsg.HCI013 <- dsg.HCI013[, colSums(dsg.HCI013) > 0] # remove non-observable coefficients
```

```
limma::makeContrasts(PT.FulvesVsControl = (mouseP7M21 + mouseP7M22 + mouseP7M23 + mouseP7M24)/4 - (mouseP9M13 + mouseP9M14 + mouseP9M15 + mouseP9M16)/4,
            PT.ComboVsControl = (mouseP9M28 + mouseP9M29 + mouseP9M30 + mouseP9M31)/4 - (mouseP9M13 + mouseP9M14 + mouseP9M15 + mouseP9M16 )/4,
            PT.RiboVsContril = (mouseP9M21 + mouseP9M22 + mouseP9M26 + mouseP9M9)/4 - (mouseP9M13 + mouseP9M14 + mouseP9M15 + mouseP9M16 )/4,
            ###
            LUNGS.FulvesVsControl =(mouseP7M21 + mouseP7M22 + mouseP7M23 + mouseP7M24)/4 + Fulves.LUNGS
              - (mouseP9M13 + mouseP9M14 + mouseP9M15 + mouseP9M16)/4 - Control.LUNGS,
            LUNGS.ComboVsControl =(mouseP9M28 + mouseP9M29 + mouseP9M30 + mouseP9M31)/4 + Combo.LUNGS
            - (mouseP9M13 + mouseP9M14 + mouseP9M15 + mouseP9M16)/4 - Control.LUNGS,
            LUNGS.RiboVsControl =(mouseP9M21 + mouseP9M22 + mouseP9M26 + mouseP9M9)/4 + Ribo.LUNGS
            - (mouseP9M13 + mouseP9M14 + mouseP9M15 + mouseP9M16)/4 - Control.LUNGS,
```

edgeR approach

- remove redundancy ↔ treatments can be reconstructed from mouse covariates

```
#between and within subject comparisons
dsg.HCI013 <- model.matrix(~ 0 + mouse, data = dge.HCI013$samples)  #start with mouse effect
#site specific treatments, except PT, this is the reference
#table(dge.HCI013$samples$SampleGroup,dge.HCI013$samples$site)
Control.LUNGS <- dge.HCI013$samples$SampleGroup == "Control" & dge.HCI013$samples$site == "LUNGS"
Ribo.LUNGS <- dge.HCI013$samples$SampleGroup == "Ribo" & dge.HCI013$samples$site == "LUNGS"
Combo.LIVER <- dge.HCI013$samples$SampleGroup == "Combo" & dge.HCI013$samples$site == "LIVER"
Combo.LUNGS <- dge.HCI013$samples$SampleGroup == "Combo" & dge.HCI013$samples$site == "LUNGS"
Combo.KIDNEY <- dge.HCI013$samples$SampleGroup == "Combo" & dge.HCI013$samples$site == "KIDNEY"
Fulves.LUNGS <- dge.HCI013$samples$SampleGroup == "Fulves" & dge.HCI013$samples$site == "LUNGS"
Fulves.LIVER <- dge.HCI013$samples$SampleGroup == "Fulves" & dge.HCI013$samples$site == "LIVER"
Fulves.KIDNEY <- dge.HCI013$samples$SampleGroup == "Fulves" & dge.HCI013$samples$site == "KIDNEY"

dsg.HCI013 <- cbind(dsg.HCI013,Control.LUNGS,Ribo.LUNGS,Combo.LIVER,Combo.LUNGS,Combo.KIDNEY,Fulves.LUNGS,Fulves.LIVER,Fulves.KIDNEY)
dsg.HCI013 <- dsg.HCI013[, colSums(dsg.HCI013) > 0] # remove non-observable coefficients
```

```
limma::makeContrasts(PT.FulvesVsControl = (mouseP7M21 + mouseP7M22 + mouseP7M23 + mouseP7M24)/4 - (mouseP9M13 + mouseP9M14 + mouseP9M15 + mouseP9M16)/4,
            PT.ComboVsControl = (mouseP9M28 + mouseP9M29 + mouseP9M30 + mouseP9M31)/4 - (mouseP9M13 + mouseP9M14 + mouseP9M15 + mouseP9M16 )/4,
            PT.RiboVsContril = (mouseP9M21 + mouseP9M22 + mouseP9M26 + mouseP9M9)/4 - (mouseP9M13 + mouseP9M14 + mouseP9M15 + mouseP9M16 )/4,
            ###
            LUNGS.FulvesVsControl =(mouseP7M21 + mouseP7M22 + mouseP7M23 + mouseP7M24)/4 + Fulves.LUNGS
              - (mouseP9M13 + mouseP9M14 + mouseP9M15 + mouseP9M16)/4 - Control.LUNGS,
            LUNGS.ComboVsControl =(mouseP9M28 + mouseP9M29 + mouseP9M30 + mouseP9M31)/4 + Combo.LUNGS
            - (mouseP9M13 + mouseP9M14 + mouseP9M15 + mouseP9M16)/4 - Control.LUNGS,
            LUNGS.RiboVsControl =(mouseP9M21 + mouseP9M22 + mouseP9M26 + mouseP9M9)/4 + Ribo.LUNGS
            - (mouseP9M13 + mouseP9M14 + mouseP9M15 + mouseP9M16)/4 - Control.LUNGS,
```

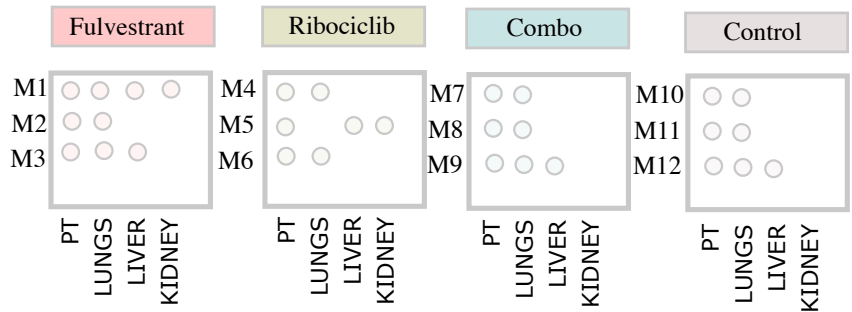
```
#GLM fit
edge.fit.HCI013 <- glmQLFit(dge.HCI013, dsg.HCI013)
```

edgeR approach

- Fixed-effect model (edgeR does not support random effect models)
- Standard edgeR workflow can be used.
- Creating the design matrix and proper contrasts can be challenging, though "algorithmic"
- Observation: Non-significant "tails" in volcano plots (big average effect but $p.adj \approx 1$)

Treat mouse variables as random effects

patient material (initial tumor) grown in mice: patient-derived xenographs (PDXs)



covariates: *treatment*, *site*, **mouse** $\leftrightarrow$ paired structure expressed via correlations

fixed effects: an explanatory variable whose relationship to the response variable is constant (fixed) across all observations

random effects: explanatory variable is random, its relationship to the response variable varies across observations.

limma-voom framework

- For (standard) fixed-effect models

$$y_g = X\beta_g + \epsilon_g, \qquad \epsilon_g \sim \mathsf{N}(0, \mathrm{diag}(w_g)\sigma_g^2) \tag{1}$$

y_g : $\log_2$ CPM , X : design matrix, β_g : regression coefs, ϵ_g : residuals with variances σ_g^2, w_g : precision weights learnt by voom

limma-voom framework

- For (standard) fixed-effect models

$$y_g = X\beta_g + \epsilon_g, \qquad \epsilon_g \sim \mathsf{N}(0, \mathrm{diag}(w_g)\sigma_g^2) \tag{1}$$

y_g : $\log_2$ CPM , X : design matrix, β_g : regression coefs, ϵ_g : residuals with variances σ_g^2, w_g : precision weights learnt by voom

- Repeated measuments: linear mixed effect model with a *single* random effect

$$y_g = X\beta_g + Z\alpha_g + \epsilon_g, \qquad \alpha_g \sim \mathsf{N}(0, \tau_g^2),\ \epsilon_g \sim \mathsf{N}(0, \mathrm{diag}(w_g)\sigma_g^2), \tag{2}$$

limma-voom framework

- two-step implementation using duplicateCorrelation () function
  1. Fit the model (2) using REML (restricted maximimum likelihood estimation) for each gene, infer τ_g ($\approx$ intraclass correlation),
     Define a single genome-wide variance term

$$\tau^2 = \tanh\left(\frac{1}{G}\sum_{g=1}^{G} \operatorname{arctanh} \tau_g^2\right)$$

  2. Fit model of a type (1)

$$y_g = X\beta_g + \epsilon_g, \quad \epsilon_g \sim \mathsf{N}(0, \operatorname{diag}(w_g)\Sigma_\epsilon), \quad \Sigma_\epsilon = \begin{pmatrix} 1 & \tau^2 & & \\ \tau^2 & 1 & & \\ & & 1 & \\ & & & \ddots \end{pmatrix}$$

limma-voom framework

```
# apply duplicateCorrelation is two rounds
design = model.matrix( ~ Disease, metadata)
vobj_tmp = voom( geneExpr, design, plot=FALSE)
dupcor <- duplicateCorrelation(vobj_tmp,design,block=metadata$Individual)

# run voom considering the duplicateCorrelation results
# in order to compute more accurate precision weights
# Otherwise, use the results from the first voom run
vobj = voom( geneExpr, design, plot=FALSE, block=metadata$Individual, correlation=dupcor$consensus)

# Estimate linear mixed model with a single variance component
# Fit the model for each gene,
dupcor <- duplicateCorrelation(vobj, design, block=metadata$Individual)

# But this step uses only the genome-wide average for the random effect
fitDupCor <- lmFit(vobj, design, block=metadata$Individual, correlation=dupcor$consensus)

# Fit Empirical Bayes for moderated t-statistics
fitDupCor <- eBayes( fitDupCor )
```

limma-voom framework

```
design <- model.matrix(~ 0 + treatment.site, data = dge.HCI013$samples)
colnames(design) <- str_remove(colnames(design),"treatment.site")

my.contrast <- limma::makeContrasts(PT.FulvesVsControl = Fulves.PT - Control.PT,
                                    PT.ComboVsControl =  Combo.PT - Control.PT,
                                    PT.RiboVsControl = Ribo.PT - Control.PT,
                                    ###
                                    LUNGS.FulvesVsControl = Fulves.LUNGS - Control.LUNGS,
                                    LUNGS.ComboVsControl = Combo.LUNGS - Control.LUNGS,
                                    LUNGS.RiboVsControl = Ribo.LUNGS - Control.LUNGS,
                                    # within treatment, site vs PT
                                    #Fulvestrant
                                    Fulves.LUNGSvsPT = Fulves.LUNGS - Fulves.PT,
                                    Fulves.LIVERvsPT = Fulves.LIVER - Fulves.PT,
                                    Fulves.KIDNEYvsPT = Fulves.KIDNEY - Fulves.PT,
                                    #Combo
                                    Combo.LUNGSvsPT = Combo.LUNGS - Combo.PT,
                                    Combo.LIVERvsPT = Combo.LIVER - Combo.PT,
                                    Combo.KIDNEYvsPT = Combo.KIDNEY - Combo.PT,
                                    #Ribociclib alone
                                    Ribo.LUNGSvsPT = Ribo.LUNGS - Ribo.PT,
                                    #Controls alone
                                    Control.LUNGSvsPT = Control.LUNGS - Control.PT,
                                    #
                                    #doubleContrast
                                    DifEffect.Fulves.LUNGSvsPT =  (Fulves.LUNGS - Control.LUNGS) - (Fulves.PT - Control.PT),
                                    DifEffect.Combo.LUNGSvsPT =  (Combo.LUNGS - Control.LUNGS) - (Combo.PT - Control.PT),
                                    DifEffect.Ribo.LUNGSvsPT =  (Ribo.LUNGS - Control.LUNGS) - (Ribo.PT - Control.PT),
                                    levels = design)
```

```
pdf("plots/voom.pdf")
fit <- voomLmFit(dge.HCI013, design, block=dge.HCI013$samples$mouse, plot=TRUE)
dev.off()
#First intra-block correlation  0.1287046
#Final intra-block correlation  0.1288309

fit2 <- contrasts.fit(fit, my.contrast)
fit2 <- eBayes(fit2)
```

limma approach

- One random effect implemented and merged with limma-voom framework through one function voomLmFit()
- Estimates a global whole-genome correlation, time efficient though might inflate FDR
- Easy to write down contrasts

Dream framework

Hoffman, Gabriel E., and Panos Roussos. "Dream: powerful differential expression analysis for repeated measures designs." Bioinformatics 37.2 (2021),

- complete mixed-effect approach (gene-specific correlations, multiple random effects)

$$y_g = X\beta_g + \sum_j Z_j \alpha_g^{(j)} + \epsilon_g, \quad \alpha_g \sim \mathsf{N}(0, \tau_{g,j}^2),\ \epsilon_g \sim \mathsf{N}(0, \mathrm{diag}(w_g)\sigma_g^2) \quad \rightsquigarrow \quad \hat{\beta}, \hat{\tau}_{g,j},$$

- compatible with limma-voom framework
- more resource hungry

The dream method replaces 4 core functions of limma with a linear mixed model.

1. `voomWithDreamWeights()` replaces `voom()` to estimate precision weights
2. `dream()` replaces `lmFit()` to estimate regression coefficients.
3. `variancePartition::eBayes()` replaces `limma::eBayes()` to apply empircal Bayes shrinkage on linear mixed models.
4. `variancePartition::topTable()` replaces `limma::topTable()` to give seamless access to results from `dream().

Dream framework

```
# The variable to be tested must be a fixed effect
form <- ~ 0 + treatment.site + (1|mouse)

#Contrasts
L2 = makeContrastsDream( form, dge.HCI013$samples, contrasts =
                           c(PT.FulvesvsControl = "treatment.siteFulves.PT - treatment.siteControl.PT",
                             PT.ComboVsControl =  "treatment.siteCombo.PT - treatment.siteControl.PT",
                             PT.RiboVsControl = "treatment.siteRibo.PT - treatment.siteControl.PT",
                             #####
                             LUNGS.FulvesVsControl = "treatment.siteFulves.LUNGS - treatment.siteControl.LUNGS",
                             LUNGS.ComboVsControl = "treatment.siteCombo.LUNGS - treatment.siteControl.LUNGS",
                             LUNGS.RiboVsControl = "treatment.siteRibo.LUNGS - treatment.siteControl.LUNGS",
                             # within treatment, site vs PT
                             #Fulvestrant
                             Fulves.LUNGSvsPT = "treatment.siteFulves.LUNGS - treatment.siteFulves.PT",
                             Fulves.LIVERvsPT = "treatment.siteFulves.LIVER - treatment.siteFulves.PT",
                             Fulves.KIDNEYvsPT = "treatment.siteFulves.KIDNEY - treatment.siteFulves.PT",
                             #Combo
                             Combo.LUNGSvsPT = "treatment.siteCombo.LUNGS - treatment.siteCombo.PT",
                             Combo.LIVERvsPT = "treatment.siteCombo.LIVER - treatment.siteCombo.PT",
                             Combo.KIDNEYvsPT = "treatment.siteCombo.KIDNEY - treatment.siteCombo.PT",
                             #Ribociclib alone
                             Ribo.LUNGSvsPT = "treatment.siteRibo.LUNGS - treatment.siteRibo.PT"))
```

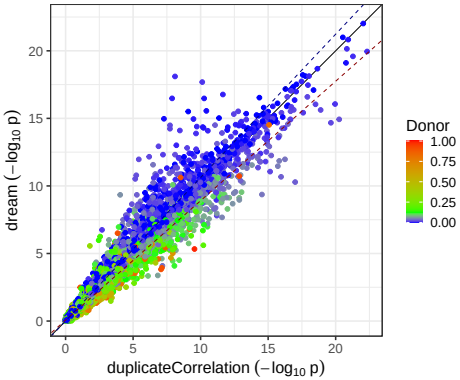
```
# estimate weights using linear mixed model of dream
set.seed(123)
vobjDream = voomWithDreamWeights( dge.HCI013, form, dge.HCI013$samples, BPPARAM=param ) #Total:121 s

# Fit the dream model on each gene
# By default, uses the Satterthwaite approximation for the hypothesis test
#The design comes here!!
set.seed(101)
fitmm = dream( vobjDream, form, dge.HCI013$samples, BPPARAM = param,  L = L2) #Total:877 s
fitmm2 = eBayes(fitmm)
```

Dream approach

- Complete fixed-effect modelling machinery
- More time consuming
- Easy to write down contrasts

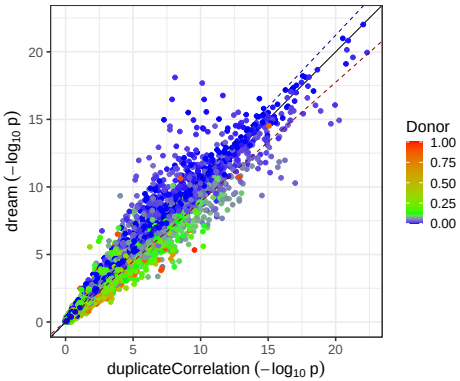
Compare p-values from dream and duplicateCorrelation

my data, quite consistent results

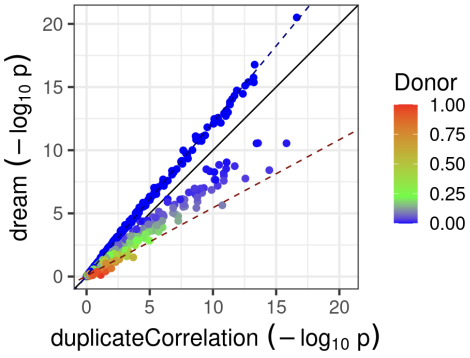


Compare p-values from dream and duplicateCorrelation

my data, quite consistent results

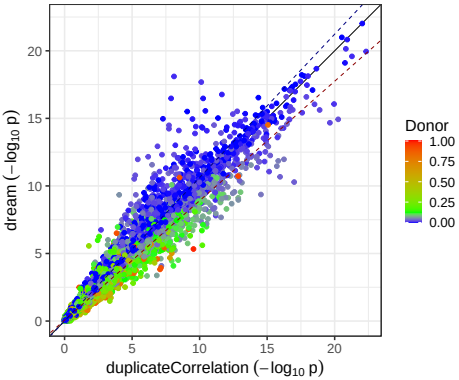


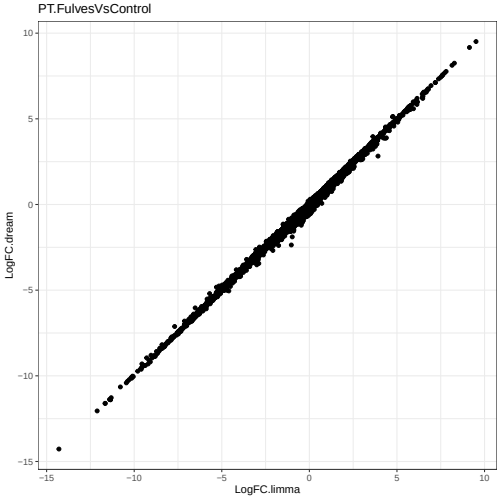
example data, not so consistent



Compare p-values from dream and duplicateCorrelation

my data, quite consistent results





Summary

- Multiple approaches combine different levels of complexity, always a trade-off
- My choice for the project: limma with block design
- Dream:
  - nicely implemented into limma environment
  - for my data, seems like an "overkill"
  - detailed analysis of variance distribution among different covariates (interpretation???)
  - some steps would require more clarification in the vignette.